

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
Please do not report the images to the
Image Problem Mailbox.

The Product: Vicinity Service

Cobrow finds other persons browsing the same page or pages in your vicinity. It transforms the web into a virtual space where people meet on web pages. No more dedicated forums and chat rooms. You can meet on every page.

Many web users are browsing the same pages at the same time, however they do not notice each other. Not because they ignore the others, but because they cannot see anybody. Cobrow makes the other users visible, it shows the icons and names of those in the vicinity.

Cobrow's impact is an improved user experience of online newspapers, virtual stores and many other types of websites. Customers will talk to each other. Sales persons will notice customers entering the virtual store. Content providers will get really close to their customers.

A text based chat is already integrated. Internet telephony can be started by a simple click on the other person's icon.

Cobrow extends WWW servers. The software is added to existing web sites. The document database remains untouched. Cobrow does not require software installation at clients.

[[System requirements and compatibility](#)]
[[Advantages for users and web sites](#)]
[[Background information](#)]

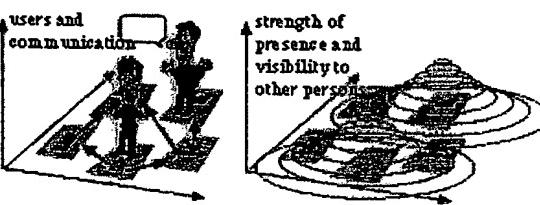


CoBrow Details

Vicinity and Metrics

The term vicinity is at the core of the CoBrow service. It describes the fact that several users are temporarily close to each other in the Web. The degree of closeness is measured with metrics. One such metric is the number of links between the pages viewed by two users at the same time. If the users are on the same page, their distance is 0, if one or the other would have to follow 2 links to be on the same page, their distance is 2. Other metrics are time or document content.

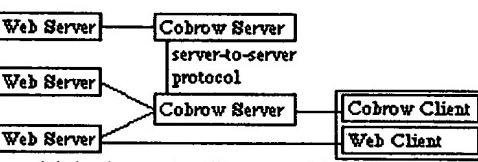
The Web is the base infrastructure of the virtual world. It is depicted as 2 dimensional hyperspace although it is actually a graph. The left side shows a simplistic representation of two persons at different locations. The right side shows the values of the persons' visibility functions. Persons see each other if the visibility overlaps.



The CoBrow System

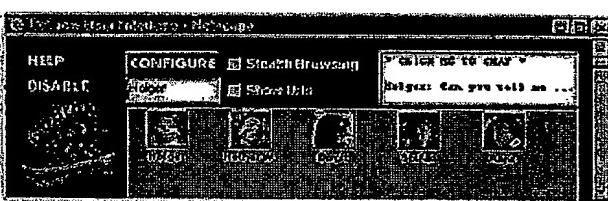
CoBrow consists of many functional entities. The core functionality is built on the Vicinity Evaluator, also referred to as the CoBrow server, which is typically co-located with a WWW-server. Based on location information, it calculates the vicinity of all users according to one or more metrics. The Vicinity Evaluator is probably the most complex CoBrow component. This is not only because it has to deal with a vast amount of data. There will be many Vicinity Evaluators in the Internet - CoBrow is a fully distributed service.

All Vicinity Evaluators communicate with each other to exchange information on Web users using an http-compliant protocol. The inter-Vicinity Evaluator communication is entirely transparent to the CoBrow clients. For the client software it is only one CoBrow service which is actually provided by many interactive Vicinity Evaluators.



The vicinity visualization is presented to the user by CoBrow clients. Today CoBrow clients are implemented as Java-applets, downloaded into the browser software as soon as a user enters a CoBrow-enabled server.

These clients serve as user interfaces not only to show the user where he is and who his neighbors are, but also to configure the vicinity calculation, and to start external communication tools. A VRML-based client with advanced vicinity visualization techniques is under development.



Synchronous Communication tools can be started and controlled from CoBrow. They are however not an integral component of CoBrow. CoBrow is able to use many different synchronous communication tools such as the MBone tools, the WebMedia toolkit, and CUSeeMe. An interface to the plain old telephone system and ISDN is under development.

Applications

[[Summary of this section](#)]

Virtual Shopping

Cobrow improves the experience of users of virtual stores. Customers see each other, and they can talk to each other. Sales persons notice customers entering a virtual store. They can talk directly to the customers while they are there. They do not have to wait for customers to reply via email or HTML-forms based feedback. See the [Showcase Web site](#).



Replacement for Chat Rooms

Cobrow makes any page a chat room. It renders dedicated chat pages, chat rooms, and chat tools superfluous. Users are able to talk to each other on any page. Users are made visible to each other on any page. The visible area can span multiple web sites.

CSCW

Cobrow is an enhancement for asynchronous computer supported collaborative work based on Web interfaces. Cobrow augments Web based CSCW services with direct visibility and synchronous communication. Users of asynchronous CSCW see other users online. The CSCW service is turned into a living virtual work space

Chance Meetings on the Web

Cobrow is useful on every Web page, on every Web site. It makes people aware of others browsing the same pages. People bump into each other on a Cobrow enabled web site like they do in the real world: street corners, shops, stores, cafes, etc. Users (e.g. authors of scientific publications) can be virtually present on certain pages permanently. People browsing by will notice them and will be able to talk to them.

User Tracking

The user tracking component is useful for browser tracking even without a neighborhood display for the user. Web site operators can watch people moving from page to page in a 3d site viewer.

Web Browser Loneliness

[more...](#)

The Internet has become an indispensable database of knowledge and information. And the World Wide Web (Web) is the most popular tool for information retrieval from the Internet. It supports asynchronous communication, particularly information dissemination based on documents of various media types. Although synchronous communication is not supported by the Web's native mechanisms, it has been integrated by means of protocol handlers, native or interpreted plug-ins, and other extension methods in Web clients.

Millions of people are browsing at the same time, sometimes hundreds or thousands the same Web site concurrently. However, people browsing the Web are still unaware of other fellow browsers. Everyone is browsing individually. There are no indications of other people, other than increased delay and congestion at certain times of the day. Browsing the Web is like shopping downtown without people on the street. There are not even sales people in the virtual shops and department stores. If it is good or bad to meet people on the street is left to the opinion of the reader. Nevertheless most humans want to meet others. And it is doubtful that online stores can afford not to talk to the potential customer in the long term.

The Web appears as mesh of documents, seemingly lifeless and static. This actually is wrong, there are many people around. Some are moving very fast from page to page. Others are walking slowly. People go on the Web to work, to find information, or to have fun. Others are just looking around without a certain destination like window shopping. There are even unmanned vehicles - robots - on the streets to pick up information. Thus there is plenty of life. But it is unfortunately hidden from users.

The real world analogy gives a strong indication that an important component - the user dimension - is missing on the Web. Adding the user dimension turns the Web into a real cyberspace.

[more...](#)

Previous Solutions

[more...](#)

Dynamic Directory Services

A straight-forward approach to notify users of others online is to provide dynamic directory services (DDS, see e.g. [Firefly][Mirabilis]). DDSs allow people to create so-called buddy-lists with friends or associates. The DDS server notifies each registered user when someone on their buddy-list enters the Web. DDSs give the user a glimpse of the Web's vivid nature, but they are not flexible enough to enable accidental meetings, e.g. of people who are interested in similar topics or are just at the same place at the same time. Thus one will never meet new partners or friends in cyberspace. Additionally, all participating users have to install special software - the DDS client.

Static Neighborhoods and Communities

Web communities or virtual meeting rooms (VMR) provide locations where people can meet in cyberspace. A VMR typically consists of a set of Web pages. Users browsing these pages see all others browsing the pages of the same VMR. Since the set of pages of a VMR is usually static and has fixed boundaries, the users are in a static neighborhood. Unlike DDSs, static neighborhoods allow people to meet others they have never met before. The user specifies the location of the VMR and not the users themselves in buddy-lists. The VMR computes the list of neighbors.

There are many different ways to implement VMRs, from static HTML-pages and chat-rooms to complex database systems. Even dynamic directory services can be used to simulate virtual meeting rooms if the system generates the buddy-lists for all participants dynamically.

A major characteristic of VMRs is the static definition as a set of pages at a fixed location. VMRs are like real conference rooms or conference centers. People go to a certain location, and they are guaranteed to meet other persons there, providing the others entered the same VMR. All persons in the VMR are visible to each other. The visible region - the set of pages - is the same for all users of a VMR. It is defined by the configuration of the VMR. The visible region is independent from the user's actual position within the VMR.

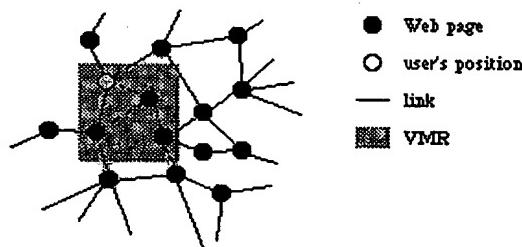


Figure 1: Example for a virtual meeting room composed of Web pages.

VMRs model closed rooms. This is useful for activities, which are best conducted in closed groups or closed rooms like synchronous or asynchronous collaborative work, meetings or lectures. However such a model is not suited for many other activities on the Web like browsing, window shopping, and individual hunt for information.

[more...](#)

The Cobrow Solution

[more...](#)

Dynamic Neighborhood

Dynamic neighborhoods are a more generic model. We will show that dynamic neighborhoods are better suited to represent the dynamic nature of the Web, and that the model is more general in the sense that it includes static neighborhoods and virtual meeting rooms. This chapter introduces the notion of a dynamic neighborhood and motivates it. The next chapter elaborates on the service model as a foundation of the implementation. The architecture of our implementation is described in chapter 3. Chapter 4 discusses results and a chapter on our current and future work concludes this paper.

In the real world (outside of closed rooms), the visible area is different from person to person. Everyone has an individual view of the world. As a result, every person sees a different subset of the objects in his neighborhood, e.g. other people, things, pictures, etc. This means to a Web neighborhood model that every Web user needs his own personal visible area, and a personal set of objects within the visible area. The visible area must not only depend on the surroundings - the Web pages. It should also correlate to the user and its properties. Whereas the static neighborhood is exclusively based on the environment, the dynamic neighborhood takes the user into account. In a static neighborhoods the visibility is the same for each user and pre-set by the VMR. In the dynamic neighborhood the visible area is tied to the user, and computed for each user individually with changing parameters.

The most obvious property of a Web user is of course the location of Web pages visited. Like in the real world, the position of a person determines the visible area. But the surroundings are also influential. On the Web, the environment is given by Web pages and interconnections (hypertext references). As people move from page to page they carry their neighborhood with them. And at the same time the neighborhood changes with every move. In the simplest model (as shown in figure 2) the neighborhood consists of the Web pages, which are close to the current page in terms of links. The user is able to see the neighboring pages and all objects associated with these pages, i.e. other users. A moving user is presented with a changing set of visible objects. Other users may see the user in motion appearing in, and disappearing from their neighborhood.

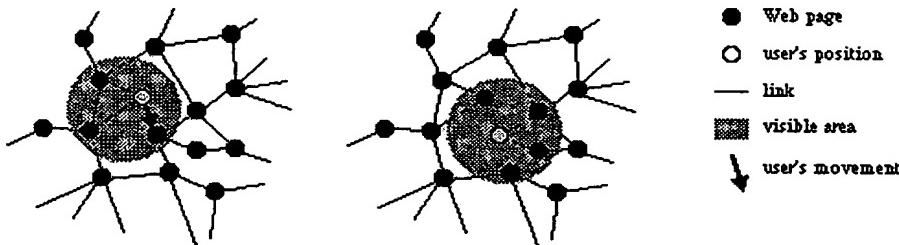


Figure 2: Example for a Web user moving from one page to another. The neighborhood is tied to the user. It moves with the user and changes.

Other than in the real world paradigm, there is much more freedom in the definition of the individual neighborhood in Web. Abstract static neighborhoods which are defined by other parameters than plain Web locations (URLs) have already been implemented. One example is the skills and interest based neighborhood of WerWeissWas service (WhoKnowsWhat) [WerWeissWas]. This neighborhood is user oriented. It does not depend on Web locations at all, neither static nor dynamic ones. A neighborhood is defined by similarity in personal skills and interests.

Later in the paper we will identify many factors and parameters for the computation of dynamic neighborhoods. The purpose of all neighborhood models is to relate people with each other in a way that increases the usefulness of the Web. The neighborhood service has to guess which association might be useful to users. To improve the guess it will acquire movement data (e.g. track users), evaluate visited documents, and ask the user for explicit information (skills, interests).

The Web - URLs and links - constitutes the infrastructure and a major part of the content. Other objects are not visible via the Web's native mechanisms. An additional service - the equivalent of night-vision goggles - is necessary to make these hidden objects visible to browsers.

[more...](#)

Service Model

[more...](#)

Locations

We believe that the infrastructure given by the World Wide Web is a solid foundation for the virtual world we are going to model. Addresses and links of the Web are based on Universal Resource Locators (URL). URLs and their superset Universal Resource Names (URN [RFC2141]) will probably represent locations on the Web for a long time to come. We will use the term URL, but all concepts can be applied equally to URNs as soon as they are being used on the Web.

We are regarding Web pages, and all other types of network accessible documents, as locations in the virtual world. These virtual locations correspond to places in the real world like rooms, street corners, and stores. People are moving - browsing - between virtual locations via hypertext references. Hypertext references (called links) are the connections between locations. They correspond to streets and paths in the real world. The only difference is that it is not possible to stop on the street until the destination is reached.

Locations do not necessarily require static HTML pages. There are many abstract locations imaginable, e.g. those mentioned above, which are based on user skills and interests. These locations are not part of the URL space. However, they can be represented by URLs for the purpose of a dynamic neighborhood model. Their URLs consist of the address of the service provider or the database server and the user's parameters (e.g. coded interests).

Links

Links between locations are currently used only as interconnections. But they have other important features. The rel-attribute of HTML-links is an example of a link attribute which contains meta-information. We propose - and use in our implementation - an additional distance-attribute to the anchor-tag of HTML to indicate the strength of interconnection. Locations can be far apart in the virtual world even if they are linked with a hypertext reference. This is useful to separate documents, which are linked but not related to each other. Examples are hotlist pages (or bookmark pages), which link to many unrelated documents. The opposite case is a distance of 0. A 0-distance combines the linked documents to a single virtual one. The distance-attribute defaults to 1 if absent, which of course is the normal case for unmodified Web sites. The distance-attribute either contains numerical values in units of hypertext references (float values), or a named relationship. Currently, defined named relationships are "near", "far", and "unrelated".

In most cases there is only a uni-directional hypertext reference between pages. This is the reason for the well known "dangling links"-problem (e.g. see [Fielding94]). We decided to treat links between documents as symmetric to allow for symmetric visibility. This decision is not part of the neighborhood model and it can be changed in a different implementation.

Meeting Rooms

The "unrelated"-distance-attribute mentioned above indicates complete separation of locations like a wall between rooms. While people can move easily between pages by clicking on the hypertext reference, they cannot see what's behind the wall. Essentially the "unrelated"-marked anchor tags in HTML documents are ignored.

This feature is useful to create virtual meeting rooms and closed groups. VMRs on the Web are represented by URLs and documents. They can be linked into the Web via hypertext references like any other Web page. Access control is provided by the Web's native mechanisms. If references to the VMR's documents are marked as "unrelated", they are like solid walls around a meeting room. This shows that the distance of links is not necessarily symmetric.

Static virtual meeting rooms fit very well into the dynamic neighborhood. They are just a special case. The configuration of Web pages of a VMR is such that all persons in the VMR have the same view - the same visible area.

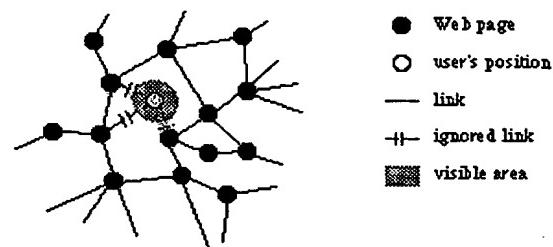


Figure 3: The visible area is restricted by links, which are marked with an "unrelated"-distance. The page is isolated from the Web. Isolation means that it is not possible to watch the surroundings from the page and vice versa.

Persons and Communication

The virtual world contains more than just locations and links. Humans, and other active entities (e.g. robots) are acting in the space constituted by URLs. We identified two important types of actions. The first is movement through the virtual world, as it has been described above. The second action is communication between active entities, e.g. humans or agents. In the real world we do not only see other persons in the neighborhood, but we communicate via various means, or we at least notice communication between people.

Both persons and communication add a new dimension to the Web. Whereas the Web serves as the environment, persons and communication exist within the space created by the Web. Of course, persons are not present everywhere while they are browsing the Web. The presence is limited to their position and its surroundings. The strength of presence - or visibility - of persons depends on their visibility-function. The visibility-function is a function of the distance from the person's position. The distance is measured in a metric imposed on the underlying hyperspace, the Web. The obvious choice as a metric are the hypertext references as described above, but others such as document content overlap are imaginable.

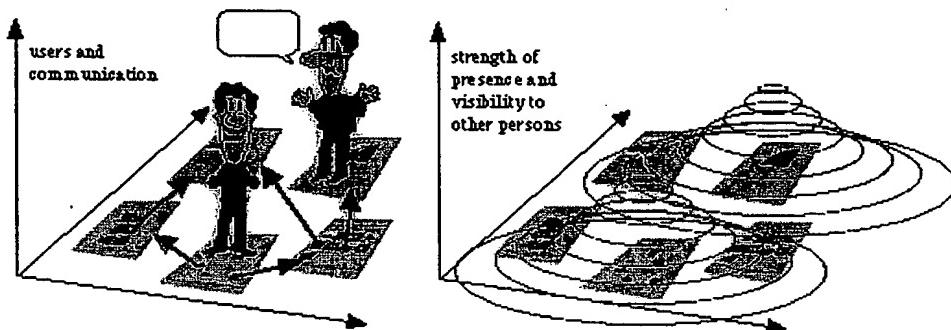


Figure 4: The Web is the base infrastructure of the virtual world. Other objects add new dimensions. The Web is depicted as 2 dimensional hyperspace although it is actually a graph. The left side shows a simplistic representation of 2 persons at different locations. The right side shows the values of the persons' visibility-functions. People can communicate if the visibility overlaps.

Communication between persons can be established if their visibility-functions overlap. The communication itself is carried out by synchronous services such as the MBONE tools or Internet telephony. The communication quality is beyond the scope of the neighborhood model, i.e. the quality is not degraded for small overlap of visibility functions as opposed to the real world.

Communication is represented by communication objects. These objects fit well into the dynamic neighborhood model. Communication objects - like persons - have a visibility-function. The function depends on the properties of the objects involved. The communication is visible to a third person, if the visibility and the visibility of the communication objects overlap. Of course there are provisions for privacy, if this is

desired.

Communication objects serve as coordinating instances, e.g. a communication object for MBONÉ tools allocates multicast addresses, and distributes the respective URL to the user interfaces. Communication objects for point-to-point communication tools distribute the communication URLs of the user's tools, e.g. RTSP-URLs.

[more...](#)

Service Abstraction

[more...](#)

Objects

The dynamic neighborhood model is based on an extensible list of object types (classes in OOP-notation, but we are not going to paint our model object-oriented. The ideas presented here are independent of the programming model). The current implementation covers 3 types of objects:

- locations
- persons, and
- communication.

Each instance - including locations - has a visibility-function. The function represents the strength of an object's presence in the space created by URLs.

Object Identification

Objects are identified by type and name. This name is globally unique for each object type. The name can be tailored to the object type for better readability and to simplify addressing of external resources. Locations are the most important resource external to the dynamic neighborhood model. The concept of URLs and URNs is imported from the Web. URLs are used as names of Location objects.

Person objects and communication objects are created by the components of the dynamic neighborhood service. Their names are constructed unique by including the host name of the dynamic neighborhood server, which creates the object. An example for a person's identifier is p123@servicehost.domain:serviceport. Implementations may use other globally unique ASCII-strings, e.g. email addresses of users, if they are known to the service.

Variables

Each object has instance variables to store individual information about the object. Part of the information such as the name is entered by users. Some information is extracted by various means and the third part is computed by the service at run-time. The service specification defines a data type for each of the variables.

The most important variables of the persons are: icon, nickname, locations, and neighbors. The variables "icon" and "nickname" are used to represent persons at the user interface level. Other variables control the computation of the visible area and the visibility-function. The "neighbors"-variable is computed by the service frequently. It contains the all-important list of visible persons in the neighborhood.

Location objects have the variables icon, links and persons. The icon is used to represent the page in an advanced user interface as a thumbnail image. The "links"-variable contains a list of hypertext references of the document. It also contains properties of the links, e.g. distance. The "persons"-variable is the counterpart to the "locations"-variable of persons. This does not necessarily mean that the information is stored twice.

Methods

Objects and variables are accessed through a set of methods. Available methods are:

- GET: retrieve the content of a variable,
- PUT: assign data to a variable,
- ADD and DELETE: add and remove data partially,
- SUBSCRIBE: subscribe for notifications of changes of a variable's content,
- UNSUBSCRIBE: end a subscription,
- ASSOCIATE and DISSOCIATE: see below.

Information between components of the neighborhood service is requested and exchanged via commands. Most of the commands are data-related. They manipulate or evaluate instance variables. Variable manipulation commands consist of an object, the object's name, the variable name, the method, and optional data.

Associations

The ASSOCIATE and DISSOCIATE methods work on objects. They are used to put objects into relation. The proposed neighborhood model uses directed 2-fold associations. Each association consists of 2 objects, the source- and the target-object.

ASSOCIATE-commands are directed to the target-object. They are comprised of the target-object identification, the source-object identification and attributes containing details of the association. The objects can be of any type.

For better readability we introduced aliases for associations between specific objects:

- Location-Location: LINK and UNLINK mark hypertext references between documents. This is similar to commands used by distributed hyperlink maintenance systems like [PitJon96].
- Person-Location: ENTER and LEAVE. These commands are generated on entry and exit of Web locations. The exact times are to be determined by information gathering components. Persons can be registered with more than one location.
- Person-Communication: JOIN and LEAVE let persons enter (and leave) communication channels.
- Person-Person: SHOW and HIDE let persons hide from each other selectively.

Associations work on symbolic names too. Currently used are the symbolic names "_new", "_any", and "_visible", which mean that a command creates a new object, is to be executed by all, or by all visible objects.

[more...](#)

Service Components

[more...](#)

The neighborhood service is a distributed system. Neighborhood servers are the core components of the system. Clients are the user interfaces, and auxiliary components gather information about users and the Web.

Servers

Neighborhood servers provide the user dimension of the virtual world. They serve neighborhood clients in the same sense as Web servers serve Web clients. The Web is segmented in parts, which are created by individual Web servers. A neighborhood server is responsible for the user space of a part of the Web and it will create the user space for this part. In this case each Webserver is accompanied or complemented by a neighborhood server. But a neighborhood server can also create the user space for a group of Web servers.

A neighborhood server essentially maintains 2 databases, a link database and a user database. The link database reflects the document and link structure of the augmented Web server(s). Neighborhood servers maintain backward references for all hypertext references in order to provide symmetric visibility. Links to objects on the same server (core links) are maintained easily. Links to documents on other Web servers (surface links) generate network traffic between neighborhood servers. The LINK-command (ASSOCIATE locations) is used to announce a link. This is done very rarely so that the generated traffic is only a small fraction of the HTTP traffic between Web clients and Web servers.

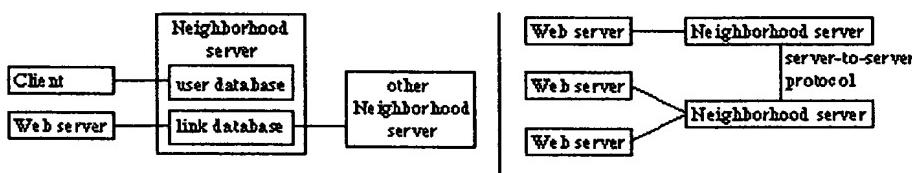


Figure 5: Left: a simplified view of a neighborhood server. It communicates with clients, Web servers, and other neighborhood servers. Right: A neighborhood server attends one or more Web servers.

The main task of neighborhood servers is the computation of the visible objects. This is done based on the Web link structure, user preferences, and various system settings. User preferences and site specific settings control the computation of the visibility-function and as such the set of visible objects. We are currently using a visibility-function box-shaped in space and time. The person's instance variable "link-distance" is the length of the box in space. This parameter is typically set to 2, which means that other persons, which are less or equal than 2 hypertext references apart from the user's position are visible (see figure 6).

The second parameter controls the depth of the visibility-function in time. The time dimension has not been shown in figure 4, but actually proved very helpful to increase the usability of the system. The time dimension of the visibility-function smoothes the movement between pages. It is much easier for the users to observe the movement if persons remain registered with locations for a short time after they left a Web page.

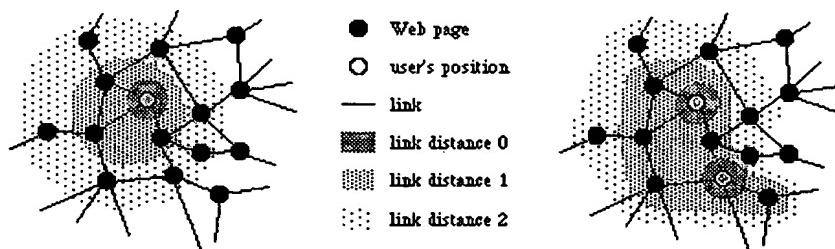


Figure 6: Pages in the visible area depending on the link-distance

variable. Right: the shape of the visibility-function can be complex, if a user is browsing multiple pages simultaneously.

The result of the computation, a list of visible pages, persons, and communications is kept available for neighborhood clients.

Clients

Neighborhood clients presents a dynamically changing view of the user space. They are using the protocol introduced above to retrieve information about the position of the user, the surroundings, other users and their properties.

Basic User Interface

The basic user interface originates from a test tool for the protocol engine and the neighborhood server. However, it is very useful as a lean user interface. It shows the neighbors as a list with icons, nicknames, and the position of the users (as URLs). The user interface allows modifications to instance variables by the user who owns the person-object in the neighborhood server.

The basic user interface is implemented in Java. The code is separated into two distinct layers. The lower layer (network layer) is a protocol handler, which implements the neighborhood protocol. We are currently using an encapsulation of the protocol in HTTP in order to tunnel through proxies and firewalls. The upper layer (display layer) consists of display classes. The display classes show the content of instance variables of specific objects of the neighborhood server, i.e names, icons and positions of the neighbors.

The display layer is independent of the protocol implementation. It will survive a major redesign of the protocol as long as the basic concepts remain the same.

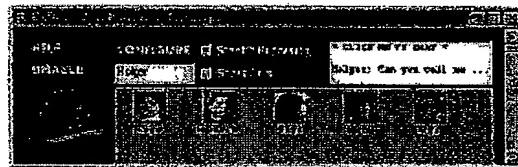


Figure 7: The basic user interface. Neighbors are shown as icons. The icons are clickable to initiate communication. The little chat-window in the upper right tries to attract people to ongoing communication.

Advanced User Interface

The VRML 2.0 standard and an external authoring interface (EAI) enable dynamic virtual reality displays. Rather than being just a list of visible persons the neighborhood can be presented as a three dimensional visualization of a (small) fraction of the Web. This is similar to [Dome1994], "A Graphical Hypertext Navigation Tool", but employs a VRML engine for the rendering of 3-dimensional scenes. The advanced user interface is using dynamic VRML objects and Java-based scripting. Web pages are represented by thumbnail images (icon instance variable of location object) with interconnections as hypertext references. Persons are shown as icons or names, which are grouped around the Web pages. Text based (chat) communication between visible persons is presented as a text track, which flows from the front to the back of the 3d-scene. Icons of persons are clickable to initiate communication as shown in the basic user interface.

[more...](#)

References

[Firefly] Firefly Network, Inc. Personalize your Network. Cambridge, MA, 1997, Software online at <URL:<http://www.firefly.net/>>.

[Mirabilis] Mirabilis Ltd, ICQ - World's Largest Internet Online Communication Network, Tel Aviv, Israel, 1997, Software online at <URL:<http://www.mirabilis.com/>>.

[WerWeissWas] WerWeissWas, a search engine for experts, Germany, 1997, Software online at <URL:<http://www.wer-weiss-was.de/>>.

[RFC2141] URN Syntax. R. Moats. Internet RFC 2141, proposed standard, May 1997, <URL:<ftp://ds.internic.net/rfc/rfc2141.txt>>.

[Fielding94] R. Fielding: Maintaining Distributed Hypertext Infostructures: Welcome to MOMspider's Web, Computer Networks and ISDN systems, Volume 27, issue 2, 1994.

[PitJon96] James E. Pitkow, R. Kipp Jones: Supporting the Web: A Distributed Hyperlink Database System, Fifth International World Wide Web Conference, Paris, France, May 6-10, 1996.

[Dome94] P. Domel: WebMap - A Graphical Hypertext Navigation Tool. Proceedings of the Second International World Wide Web Conference. Chicago: 1994. <URL:<http://www.tn.informatik.uni-frankfurt.de/~doemel/Papers/WWWFall94/www-fall94.html>>

[D4.10] EU Telematics Project CoBrow: "Compatibility assessment and integration strategy", Project Deliverable 4.10,